CLAIMS

What is claimed is:

1. A method of establishing a secure communication session among a plurality of member nodes that participate in a multicast group across a wide area network, comprising the steps of:

receiving information defining a plurality of multicast proxy service nodes that are distributed across a local area network that is coupled to the wide area network and for controlling when any of the member nodes join or leave the multicast group, wherein the multicast proxy service nodes are logically represented by a first binary tree;

creating and storing a second binary tree for representing the member nodes, wherein each of the member nodes is represented by a leaf node of the second binary tree, that is stored in a domain of a directory service that is distributed across the wide area network, and wherein each of the member nodes is capable of establishing multicast communication and serving as a key distribution center;

creating and storing a group session key associated with the multicast group and a private key associated with each node in a group using secure key exchange;

when one of the member nodes joins the multicast group, determining a new group session key by replicating a branch of the second binary tree.

2. A method as recited in Claim 1, wherein each of the member nodes is associated with one of the multicast proxy service nodes, wherein each of the multicast proxy service nodes acts as one of a plurality of replicated group controllers, further comprising the steps of:

joining one of the group controllers to the plurality of replicated group controllers in the local area network;

establishing, by one of the group controllers, a secure communication channel between one of the group controllers and another of the group controllers using a key exchange protocol;

receiving a request to add or delete the specified member node of the multicast group from a load balancer that is coupled to the plurality of group controllers;

creating and storing a new group session key for each member node in each branch of the tree that is affected by adding or deleting the specified member node from the group;

distributing the new group session key from one of the group controllers to the affected member nodes.

3. A method as recited in Claim 2, wherein distributing a group session key further comprises:

receiving a token value at the group controller to designate the group controller as having permission to selectively generate the new group session key and to generate node keys associated with the affected intermediate nodes and the leaf nodes; and

creating and storing the new group session key only when the group controller has the token value.

4. A method as recited in Claim 2, wherein distributing a group session key further comprises:

determining whether the multicast group has a node that is leaving the multicast group;

determining which of the intermediate nodes are affected by the leaving node;

updating only keys associated with the affected intermediate nodes;

generating a new group session key; and

sending the new group session key to the leaf nodes.

5. A method as recited in Claim 4, wherein updating keys comprises:

generating a new key of a parent node of the leaving node; and

encrypting the new key of the parent node with an existing key of the member node that is adjacent to the parent node.

6. A method as recited in Claim 2, wherein distributing a group session key further comprises:

receiving a request message from one of the plurality of member nodes to join the multicast group;

determining which of the intermediate nodes are affected by the joining node;

updating only keys associated with the affected intermediate nodes;

generating a new group session key and a private key for the joining node; and

sending a message comprising the new group session key, the private key, and the updated keys of affected intermediate nodes to the joining node.

7. A method as recited in Claim 6, wherein updating keys comprises performing a one way hash function on the keys associated with the affected intermediate nodes.

8. A method as recited in Claim 2, wherein establishing a secure communication channel comprises exchanging a public key of the group controller with all other group controllers in the plurality of replicated group controllers based upon optimized broadcast Diffie-Hellman protocol.

9. A method as recited in Claim 2, wherein establishing a secure communication channel comprises:

receiving a public key value that is broadcast by the joining node;

sending a collective public key value from the nodes to the joining node;

computing a shared secret key;

creating and storing a group shared secret key by exchanging private key values.

10. A method as recited in Claim 1, wherein determining a new group session key comprises computing a group shared secret key at a first member node for use in a public key process and using less than n * (n-1) messages, where "n" is a number of member nodes in a broadcast or multicast group, by the steps of:

generating an intermediate shared secret key by issuing communications to a second member node;

sending a first private value associated with the first member node to the second member node, and receiving from the second member node a second private value associated with the second member node using the intermediate shared secret key;

generating and communicating a collective public key that is based upon the first private value and the second private value to a third member node of the network;

receiving an individual public key from the third member node; and

computing and storing the group shared secret key based upon the individual public key.

11. A method as recited in Claim 10, further comprising:

joining the first member node to an initial multicast group in response to generating the intermediate shared secret key; and

joining a second member node to a new multicast group that subsumes the initial multicast group after receiving the individual public key.

12. A method as recited in Claim 10, wherein the step of communicating the collective public key further comprises determining whether the first member node or the second member node transfers the collective public key based upon an order of entry of such member nodes into a multicast group.

13. A method as recited in Claim 10, wherein sending the first private value and receiving the second private value further comprises computing the first private value as a random integer and receiving a second random integer as the second private value.

14. A method as recited in Claim 10, further comprising the step of establishing a cryptographic communication session between the first member node and the second

member node, whereby secure communications are established between the first member node and the second member node using public key exchange and only approximately 2n + 2(n-1) total messages, wherein n represents a number of the member nodes.

15. A method as recited in Claim 10, wherein generating the shared secret key value comprises computing and storing the shared secret key value "k" at the first member node according to the relation

$k = C^{ab} \bmod (q) = p^{abc} \bmod (q)$

wherein C, a, b, c, q, and p are values stored in a memory, and wherein C is the individual public key, a is the private value of the first member node, b is the private value of the second member node, c is a third private value of the third member node, p is a base value, and q is a prime number value.

16. A method as recited in Claim 1, wherein determining a new group session key comprises computing a group shared secret key, each of the member nodes having a private key value associated therewith, by the steps of:

communicating a first public key value of the first member node to a second member node;

creating and storing an initial shared secret key for the first member node and second member node based on a first private key value and a second public key value that is received from the second member node;

creating and storing information at the first member node that associates the first member node with a first network communication entity by generating a collective public key value that is shared by the first member node and a second member node and based on the first private key value and a second private key value that is derived by the first member node from the second public key value;

receiving a third public key value from a third member node that seeks to join the first network communication entity;

creating and storing a shared secret key value based on the collective public key value and the third public key value;

joining the first member node to a second network communication entity that includes the first network communication entity and the third member node and that uses secure communication with messages that are encrypted using the shared secret key value.

17. A method as recited in Claim 16, wherein joining the first member node to a second network communication entity includes the step of communicating the first private key value to the second member node and to the third member node using messages encrypted using the shared secret key value.

18. A method as recited in Claim 16, wherein creating and storing a shared secret key value further comprises creating and storing the shared secret key based upon how many times each member node of the second network communication entity has participated in formation of any such entity and based upon each private number of each member node in the second network communication entity.

19. A method as recited in Claim 16, further comprising the step of creating and storing a subsequent shared secret key for use by the first network communication entity and the third node to enable the third node to independently compute the group shared secret key.

20. A method as recited in Claim 19, wherein creating and storing the subsequent shared secret key comprises creating and storing the subsequent shared secret key, k, according to the relation

$$k = p^{(a*x)(b*y)(c*z)} \bmod (q)$$

where p = a random number, q = a prime number, a = the first private key value, b = the second private key value, c = a private key value of the third member node, x = a number of times the first member node has participated in entity formation, y = a

number of times the second member node has participated in entity formation, and z = a number of times the third member node has participated in entity formation.

21. A method as recited in Claim 16, wherein the step of joining the first member node to a second network communication entity further comprises:
creating and storing a collective public key based upon the first private key value, the second private key value, and the third private key value;
communicating a collective public key of the second network communication entity to the third member node.

22. A method as recited in Claim 16, wherein the step of joining the first member node to a second network communication entity further comprises determining which one of the member nodes of the first network communication entity is designated to transfer the collective public key based upon order of entry into the formed entity.

23. A method as recited in Claim 16, wherein creating and storing an initial shared secret key for the first member node and second member node comprises creating and storing an initial shared public key "AB" according to the relation
$AB = k_{ab}^{ab} \bmod (q) = p^{(ab)(ab)} \bmod (q)$
wherein k = the initial shared secret key value, a = the first private key value, b = the second private key value, p is a base value, and q is a randomly generated prime number value.

24. A method as recited in Claim 1, further comprising the steps of:
authenticating a first event service node with a subset of the event service nodes that are affected by an addition of the first event service node to the multicast group, based on key information stored in a directory;
receiving a plurality of private keys from the subset of nodes;
generating a new private key for the first event service node;
communicating the plurality of private keys and the new private key to the first event service node;

communicating a message to the subset of nodes that causes the subset of nodes to update their private keys.

25. A method as recited in Claim 24, wherein authenticating the plurality of event service nodes based on information stored in a directory includes authenticating the plurality of event service nodes based on a directory that comprises a directory system agent (DSA) for communicating with one or more of the event service nodes and a replication service agent (RSA) for replicating attribute information of the one or more event service nodes.

26. A method as recited in Claim 24, wherein authenticating the plurality of event service nodes based on information stored in a directory includes authenticating the plurality of event service nodes based on a directory that comprises a directory system agent (DSA) for communicating with one or more of the event service nodes and a replication service agent (RSA) for replicating attribute information of the one or more event service nodes, wherein the attribute information comprises a group session key and the private keys

27. A method as recited in Claim 24, wherein generating private keys comprises generating private keys for each of the intermediate nodes and leaf nodes, the private keys providing unique identification within the tree structure, and wherein each private key is N bits in length, wherein each bit corresponds to one of the private keys, and N is an integer.

28. A method as recited in Claim 24, further comprising distributing a group session key to all nodes by creating and storing the group session key using a first event service node of one domain of the directory; replicating the directory; and obtaining the group session key from a local event service node that is a replica of the first event service node.

29. A method as recited in Claim 24, wherein generating private keys comprises generating private keys for each of the intermediate nodes and leaf nodes, the private keys providing unique identification within the tree structure, and wherein each private key is N bits in length, wherein each bit corresponds to one of the private keys, and N is an integer; and wherein the most significant bits of the N bits correspond to nodes logically near the root node in the tree structure.

30. A method as recited in Claim 24, further comprising selectively updating a group session key for use in encrypting communications among all nodes in the multicast group and selectively updating the private keys.

31. A method as recited in Claim 30, further comprising selectively updating the group session key and the private keys by:

detecting whether a network node associated with one of the leaf nodes is leaving the secure multicast or broadcast group;

determining tree nodes along a tree path corresponding to the leaving leaf node are affected in response to the detecting step;

updating the private keys of the affected intermediate nodes;

generating a new group session key;

modifying the attribute information based upon the updated private keys and the new group session key; and

generating instructions that distribute the modified attribute information using directory replication.

32. A method as recited in Claim 31, wherein the step of generating the new group session key is performed by one of the affected intermediate nodes that is the parent node of the leaving leaf node, the one affected intermediate node selectively sending the new group session key to all ancestral nodes along the tree path.

33. A method as recited in Claim 31, wherein the step of updating the private keys of the intermediate nodes comprises performing a one way hash function on the private keys in response to receiving a corresponding instruction from the root node.

34. A method as recited in Claim 24, further comprising selectively updating a group session key and the private keys, wherein the step of selectively updating comprises:
receiving a request message from a new network node to join the secure multicast group;
determining which of the intermediate nodes along a tree path corresponding to the new leaf node are affected in response to the receiving step;
updating the private keys of the affected intermediate nodes;
generating a new group session key and a private key of the new leaf node;
modifying the attribute information based upon the updated private keys, the new group session key, and the private key of the new leaf node; and
distributing the modified attribute information using directory replication.

35. A method as recited in Claim 34, wherein one of the affected intermediate nodes that is the parent node of the new leaf node requests permission from the root node to generate the new session key.

36. A method as recited in Claim 34, wherein the step of updating the private keys of the intermediate nodes comprises performing a one way hash function on the private keys in response to receipt of a corresponding instruction from the root node.

37. A method as recited in Claim 34, wherein generating the private keys comprises generating the private keys based upon an Internet Protocol (IP) address and time values.

38. A method as recited in Claim 1, further comprising the steps of:

creating and storing a group session key associated with the multicast group in the directory service;

authenticating a first multicast proxy service node with a subset of the multicast proxy service nodes that are affected by an addition of the first multicast proxy service node to the multicast group, based on the group session key stored in the directory;

receiving a plurality of private keys from the subset of nodes;

receiving a new group session key for the multicast group, for use after addition of the first multicast proxy service node, from a local multicast proxy service node that has received the group session key through periodic replication of the directory;

communicating the new group session key private key to the first multicast proxy service node;

communicating a message to the subset of nodes that causes the subset of nodes to update their private keys.

39. A method as recited in Claim 38, wherein authenticating the plurality of multicast proxy service nodes includes authenticating the plurality of multicast proxy service nodes using a directory system agent (DSA) that communicates with one or more of the multicast proxy service nodes and a replication service agent (RSA) that replicates attribute information of the one or more multicast proxy service nodes.

40. A method as recited in Claim 38, wherein receiving a new group session key includes receiving the new group session key from a directory system agent (DSA) for communicating with one or more of the multicast proxy service nodes and a replication service agent (RSA) for replicating key information of the one or more multicast proxy service nodes.

41. A method as recited in Claim 38, further comprising the step of signaling the

replication service agent to carry out replication by storing an updated group session key in a local node of the directory.

42. A method as recited in Claim 38, further comprising distributing a group session key to all nodes by creating and storing the group session key using a first multicast proxy service node of one domain of the directory; replicating the directory; and obtaining the group session key from a local multicast proxy service node that is a replica of the first multicast proxy service node.

43. A method as recited in Claim 38, further comprising distributing a group session key to all nodes by creating and storing the group session key using a first multicast proxy service node of one domain of the directory; replicating the directory; and obtaining the group session key from a local multicast proxy service node that is a replica of the first multicast proxy service node.

44. A method as recited in Claim 38, wherein determining a new group session key comprises the steps of:

receiving information indicating that a specified node is joining the multicast group;

updating all affected keys of a subset of member nodes in a branch of the second binary tree that contains the specified joining node;

receiving a new group session key for the multicast group, for use after addition of the specified joining node, and a new private key for the specified joining node, from one of the member nodes that is local to the specified joining node;

communicating a message to the subset of member nodes that causes the subset of member nodes to update their private keys.

45. A method as recited in Claim 38, further comprising the steps of:

associating a plurality of intermediate nodes of the binary tree with a plurality of multicast service agents;

establishing a secure back channel group among the multicast service agents;

updating the group session key to all the multicast service agents by securely communicating the group session key using the secure back channel.

46. A method as recited in Claim 38, further comprising the steps of:
associating a plurality of intermediate nodes of the binary tree with a plurality of multicast service agents, wherein the multicast service agents are distributed across a wide area network;
establishing a secure back channel group among the multicast service agents;
updating the group session key to all the multicast service agents across the wide area network by securely communicating the group session key using the secure back channel.

47. A method as recited in Claim 38, further comprising the steps of:
associating a plurality of intermediate nodes of the binary tree with a plurality of multicast service agents;
establishing a secure back channel group among the multicast service agents;
updating the group session key to all the multicast service agents by securely communicating the group session key using the secure back channel;
at each intermediate node, updating the group session key of only those leaf nodes that are child nodes of the intermediate node.

48. A method as recited in Claim 38, further comprising the steps of:
receiving a request for the group session key from a publisher node that is located in a different domain from the group controller node;
determining an identifier of the publisher node using a local directory service agent;
establishing a secure communication channel among the group controller node and a directory service agent in the different domain.

49. The method as recited in Claim 38, further comprising selectively updating the group session key and the private keys by:
detecting whether a network node is leaving the secure multicast or broadcast group;

determining nodes that are affected in response to the detecting step;
updating the private keys of the affected intermediate nodes;
generating a new group session key;
modifying the attribute information based upon the updated private keys and the new group session key; and
requesting to distribute the modified attribute information using directory replication.

50. A method as recited in Claim 38, further comprising selectively updating a group session key and the private keys, wherein the step of selectively updating comprises:
receiving a request message from a new network node to join the secure multicast group;
determining which of the intermediate nodes are affected in response to the receiving step;
updating the private keys of the affected intermediate nodes;
generating a new group session key and a private key of the new node;
modifying the attribute information based upon the updated private keys, the new group session key, and the private key of the new node; and
distributing the modified attribute information to all the affected nodes.

51. A method as recited in Claim 1, further comprising managing removal of a first node from the secure multicast group that comprises the first node and a plurality of the multicast proxy service nodes, by the steps of:
creating and storing a group session key associated with the multicast group and a private key associated with each node in a directory;
receiving information indicating that the first node is leaving the multicast group;
updating all affected keys of a subset of nodes in a branch of the binary tree that contains the leaving node;
receiving a new group session key for the multicast group, for use after removal of the first node, and a new private key for the first node, from a local group controller node;

communicating a message to the subset of nodes that causes the subset of nodes to update their private keys.

52. A method as recited in Claim 51, further comprising the steps of:
associating a plurality of intermediate nodes of the binary tree with a plurality of multicast service agents;
establishing a secure back channel group among the multicast service agents;
changing a private key associated with one of the intermediate nodes, in response to receiving information indicating that the first node is leaving the multicast group;
updating a group controller of the binary tree with the changed private key by sending the changed private key by securely communicating using the secure back channel.

53. A method as recited in Claim 51, further comprising the steps of:
associating a plurality of intermediate nodes of the binary tree, including the first node, with a plurality of multicast service agents;
establishing a secure back channel group among the multicast service agents;
changing a plurality of private keys associated with all nodes that are child nodes of the first node, in response to receiving information indicating that the first node is leaving the multicast group;
updating a group controller of the binary tree with the changed private keys by decrypting the group session key from the group controller and then encrypting the group session key with the changed private keys.

54. A method as recited in Claim 51, further comprising the steps of:
associating a plurality of intermediate nodes of the binary tree with a plurality of multicast service agents;
establishing a secure back channel group among the multicast service agents;
updating the group session key to all the multicast service agents by securely communicating the group session key using the secure back channel;

at each intermediate node, updating the group session key of only those leaf nodes that are child nodes of the intermediate node.

55. A method as recited in Claim 51, further comprising the steps of:
receiving a request for the group session key from a publisher node that is located in a different domain from the group controller node;
determining an identifier of the publisher node using a local directory service agent;
establishing a secure communication channel among the group controller node and a directory service agent in the different domain.

56. A method as recited in Claim 51, further comprising distributing a group session key to all nodes by creating and storing the group session key using a first multicast proxy service node of one domain of the directory; replicating the directory; and obtaining the group session key from a local multicast proxy service node that is a replica of the first multicast proxy service node.

57. A communication system for establishing a secure communication session among a plurality of member nodes that participate in a multicast group across a wide area network, the communication system comprising:
a group controller that creates and manages secure multicast communication among the other multicast proxy service nodes, having a private key;
a computer-readable medium comprising one or more instructions which, when executed by one or more processors, cause the one or more processors to carry out the steps of:
establishing a plurality of multicast proxy service nodes that are distributed across a local area network that is coupled to the wide area network and for controlling when any of the member nodes join or leave the multicast group, wherein the multicast proxy service nodes are logically represented by a first binary tree;
creating and storing a second binary tree for representing the member nodes, wherein each of the member nodes is represented by a leaf node of the

second binary tree, that is stored in a domain of a directory service that is distributed across the wide area network, and wherein each of the member nodes is capable of establishing multicast communication and serving as a key distribution center;

creating and storing a group session key associated with the multicast group and a private key associated with each node in a group using secure key exchange;

when one of the member nodes joins the multicast group, determining a new group session key by replicating a branch of the second binary tree.

58. A computer-readable medium carrying one or more sequences of instructions for establishing a secure communication session among a plurality of member nodes that participate in a multicast group across a wide area network, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

establishing a plurality of multicast proxy service nodes that are distributed across a local area network that is coupled to the wide area network and for controlling when any of the member nodes join or leave the multicast group, wherein the multicast proxy service nodes are logically represented by a first binary tree;

creating and storing a second binary tree for representing the member nodes, wherein each of the member nodes is represented by a leaf node of the second binary tree, that is stored in a domain of a directory service that is distributed across the wide area network, and wherein each of the member nodes is capable of establishing multicast communication and serving as a key distribution center;

creating and storing a group session key associated with the multicast group and a private key associated with each node in a group using secure key exchange;

when one of the member nodes joins the multicast group, determining a new group session key by replicating a branch of the second binary tree.